

ToiletPaper #135

Enum Reverse Lookup in Kotlin

Author: Andreas Hofmann / Software Engineer / Office Stuttgart

✘ Problem

Kotlin offers the possibility to assign enumeration values to an enumeration type:

```
1 enum class Currency(val symbol: String) {  
2     DOLLAR("$"),  
3     EURO("€"),  
4     POUND("£");  
5  
6     override fun toString(): String {return this.symbol}  
7 }
```

By selecting the enumeration type, its value can be returned:

```
1 Currency.valueOf("DOLLAR") // returns the enumeration value "$"
```

But Kotlin does not offer a native function if you want to start from a value and conclude to the corresponding entry. So, you have to iterate over all enumeration types and compare the value with the searched value:

```
1 Currency.values().firstOrNull {currency -> currency.symbol == "$"} // returns the enumeration type DOLLAR
```

This kind of solution can be found searching the web in one way or another and it does what it is supposed to do. But do you want to iterate over all enumerations every time and compare values? What about a more elegant solution?

✔ Solution

We extend our enumeration class with a Companion Object:

```
1 companion object {  
2     private val mapping = values().associateBy(Currency::symbol)  
3     fun fromSymbol(symbol: String) = mapping[symbol]  
4 }
```

The function `associateBy()` returns a `map<K, T>`. More precisely a `LinkedHashMap` where `K` is the enumeration value and `T` is our enumeration type. The function `fromSymbol()` takes the enumeration value as argument and returns the matching enumeration type. Accessing the enumeration type via the enumeration value is no longer done by iterating in $O(n)$, but by a hash lookup in the static map in $O(1)$.

With this extension, the reverse lookup can be done as follows:

```
1 Currency.fromSymbol("$") // returns the enumeration type DOLLAR
```

This offers a solution with code, which is easier to read. This solution might even be more performant in terms of execution time.

Caution: If there is no 1 to 1 relation between enumeration types and enumeration values, this solution does not return the first hit but the last hit.

+ Further Aspects

- <https://kotlinlang.org/api/latest/jvm/stdlib/kotlin/enum-value-of.html>
- <https://kotlinlang.org/api/latest/jvm/stdlib/kotlin.sequences/associate-by.html>